Power and Energy Normalized Speedup Models for Heterogeneous Many Core Computing

Mohammed A. N. Al-hayanni\textsuperscript{1}, Ashur Rafiev\textsuperscript{2}, Rishad Shafik\textsuperscript{1}, Fei Xia\textsuperscript{1}

School of EEE\textsuperscript{1} and CS\textsuperscript{2}, Newcastle University
Newcastle Upon Tyne, NE1 7RU, UK
Outline

• Existing speedup models
• Motivation
• Extended heterogeneous speedup models
• Power consumption models
• Power and energy normalized speedup
• Experimental results and cross validation
• Conclusions
Amdahl’s Law

• Fixed workload
  – (50% parallelizable \( P=0.5 \))
  – On a sequential processor (single core) takes 1 unit of time to complete
Amdahl’s Law

• With two cores ...
  – Parallelizable part is distributed between the two cores
  – Total time 0.75
  – Speedup = 1/0.75 = 1.333
Amdahl’s Law

• With three cores ...
  – Speedup = 1/0.667 = 1.5
Amdahl’s Law

• With ten cores ...
  – Speedup = 1.82

\[ SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1 - P) + \frac{P}{N}} \]
Amdahl’s Law

• With \( \infty \) cores ...
  – Speedup = 2

\[
SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1 - P) + \frac{P}{N}}
\]

\[
SP(\infty) = \frac{1}{(1 - P)}
\]
Amdahl’s Law

\[ \frac{S^p}{S^1} = \left(1 - \frac{k}{N}ight) + \frac{k}{P} \]

\[ SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1 - P) + \frac{P}{N}} \]

\[ SP(\infty) = \frac{1}{(1 - P)} \]
Gustafson’s Law

• Workload scales with computing facilities
  – (50% parallelizable P=0.5)
  – On a sequential processor (single core) takes 1 unit of time to complete workload $W(1)$ designed with a single core in mind
Gustafson’s Law

• With four cores ...
  – Complete 960x540 image in the same time
  – 0.5x5=2.5 times the workload (speedup=2.5)
Gustafson’s Law

• With 16 cores ...
  – Complete FHD image in the same time
  – Speedup=0.5x17=8.5
Gustafson’s Law

• Speedup is the ratio between the improved workload and the workload before improvement
  – Calculated at fixed time

\[ W(1) = (1 - P)W + PW \]
\[ W(N) = (1 - P)W + PNW \]
\[ SP(N) = \frac{W(N)}{W(1)} = (1 - P) + PN \]
Gustafson’s Law

\[ W(\theta_k) = (1 - P)W + PW \]

\[ W(N) = (1 - P)W + PN \]

\[ SP(N) = \frac{W(N)}{W(1)} = (1 - P) + PN \]
Sun-Ni Law

• Memory-bound speedup model
  – Parallel workload per core restricted by memory structure (multi-level caches, shared memory/interfaces, etc.)
  – One core’s workload capability restricted by M – the memory of one core, N cores’ workload capability restricted by NxM
  – For the P part:

\[ W(1) = G(M) \]
\[ W(N) = G(N \times M) = G(N \times G^{-1}(W(1))) \]
Sun-Ni Law

- Multi-core speedup is derived thusly:

\[
W(N) = (1 - P)W(1) + P \times G(N \times G^{-1}(W(1)))
\]

\[
W(1) = (1 - P)W(1) + P \times G(M)
\]

\[
W(1) = (1 - P)W(1) + \frac{P \times G(N \times G^{-1}(W(1)))}{N}
\]

\[
SP(N) = \frac{W(N)}{W(1)} = \frac{(1 - P)W(1) + P \times G(N \times G^{-1}(W(1)))}{(1 - P)W(1) + \frac{P \times G(N \times G^{-1}(W(1)))}{N}}
\]
Sun-Ni Law

• Trying to remove $W(1)$:

\[
SP(N) = \frac{W(N)}{W(1)} = \frac{(1 - P)W(1) + P \times G(N \times G^{-1}(W(1)))}{(1 - P)W(1) + \frac{P \times G(N \times G^{-1}(W(1)))}{N}}
\]

If $G(x) = ax^b$ with rational $a$ and $b$

\[
G(N \times x) = N^b \times ax^b = N^b \times G(x), \text{ then}
\]

\[
G \left( N \times G^{-1}(W(1)) \right) = N^b \times G(G^{-1}(W(1))) = N^b \times W(1)
\]

\[
SP(N) = \frac{(1 - P) + P \times g(N)}{(1 - P) + \frac{P \times g(N)}{N}}, \text{ with } g(N) = N^b
\]
Sun-Ni Law

• Depending on $g(N)$
  – Sub-linear scaling (Amdahl’s if $g(N)=1$)
  – Linear scaling (Gustafson’s if $g(N)=N$)
  – Super-linear scaling (if $g(N)>N$)

• If you had more memory than cores, and the problem is memory-bound, you can scale to higher speedup than what your cores allow for compute-bound problems
Comparing the three models

**Amdahl's Law**

\[ S(k) = \frac{1}{(1 - p) + \frac{p}{k}} \]

**Gustafson's Model**

\[ S(k) = (1 - p) + pk \]

**Sun and Ni's Model**

\[ S(k) = \frac{(1 - p) + pg(k)}{(1 - p) + \frac{pg(k)}{k}} \]

\[ g(k) = k^{3/2} \]
Outline

• Existing speedup models
• Motivation
• Extended heterogeneous speedup models
• Power consumption models
• Power and energy normalized speedup
• Experimental results and cross validation
• Conclusions
Motivation

• Extend to a higher degree of core heterogeneity
• Extend to power/energy/efficiency and cover modes like dynamic voltage and frequency scaling (DVFS)
• Potential applications in run-time management systems of parallel systems
## Existing Speedup Models and the Extended Model

<table>
<thead>
<tr>
<th></th>
<th>Homogeneity</th>
<th>Heterogeneity</th>
<th>Power</th>
<th>Amdahl</th>
<th>Gustafson</th>
<th>Sun and Ni</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amdahl</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Gustafson</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sun and Ni</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Hill-Marty</td>
<td>Yes</td>
<td>Simple</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Hao and Xie</td>
<td>Yes</td>
<td>Simple</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Woo and Lee</td>
<td>Yes</td>
<td>Simple</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sun and Chen</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Extended Model</strong></td>
<td>Yes</td>
<td>Normal</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Outline

• Existing speedup models
• Motivation
• Extended heterogeneous speedup models
• Power consumption models
• Power and energy normalized speedup
• Experimental results and cross validation
• Conclusions
Heterogeneity

• Existing heterogeneity include ‘asymmetric’ and ‘dynamic’ structures (b) 😞

• We extend to cover the normal form of core heterogeneity 😊

• Still iso-ISA and not fully general 😞
Heterogeneity

- Parallel computation may not all finish together (Amdahl’s)

Diagram:

- **parallel** vs **seq.**
- **α₁**, **α₂**, **α₃**
- **no sync** vs **sync**
- **min ̄α**
BCE performance equivalence

• Calculating the performance equivalent number of BCEs
  – Based on the slowest (last to finish) core

\[ N_\alpha = \min \bar{\alpha} \cdot \sum_{i=1}^{X} N_i \]

\[ \max \bar{\alpha}, \text{avr} \bar{\alpha}, \text{opt} \bar{\alpha} \] also possible

\[ \bar{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_X\} \]
Speedup extension Amdahl’s

- Calculating the performance equivalent number of BCEs

  - Based on the slowest (last to finish) core

\[
N_\alpha = \min \bar{\alpha} \cdot \sum_{i=1}^{x} N_i
\]

\[
SP(\bar{N}) = \frac{1}{(1 - P) + \frac{P}{\alpha_x N_\alpha}}, (Amdahl's)
\]
Speedup extension Amdahl’s

- Calculating the performance equivalent number of BCEs
  - Based on the slowest (last to finish) core

\[ N_\alpha = \min \tilde{\alpha} \cdot \sum_{i=1}^{\alpha} N_i \]

\[ SP(N) = \frac{1}{(1 - P) + \frac{P}{N_\alpha}} \quad (\text{Amdahl’s}) \]

\[ SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1 - P) + \frac{P}{N}} \]

Sequential on fastest core, if \( \alpha_x \) is fastest

Parallel synced to the slowest, in case of min \( \alpha \)
Speedup extension Gustafson’s

- Gustafson’s speedup model extension
  - Again assuming sequential on a type X core and parallel on $N_\alpha$

$$SP(\bar{N}) = (1 - P)\alpha_x + PN_\alpha$$

 Sequential on fastest core, if $\alpha_x$ is fastest

 Parallel synced to the slowest, in case of min $\alpha$
Speedup extension Sun-Ni’s

- Extending Sun-Ni’s model
  - Again assuming sequential on a type $X$ core and parallel on $N_\alpha$

$$SP(\bar{N}) = \frac{(1 - P) + Pg(\bar{N})}{(1 - P)\alpha_x + \frac{Pg(\bar{N})}{N_\alpha}}$$

Sequential on fastest core, if $\alpha_x$ is fastest

Parallel synced to the slowest, in case of $\min \alpha$

Memory bound function for all cores

Reduces to extended Amdahl’s and Gustafson’s as expected
Outline

• Existing speedup models
• Motivation
• Extended heterogeneous speedup models
• Power consumption models
• Power and energy normalized speedup
• Experimental results and cross validation
• Conclusions
Power

• Divide power into effective and idle

\[ W_{total} = W(N) + W_{idle} \]

Effective power: power used by workload

\[ W(N) = \frac{W_ST_S(N) + W_PT_P(N)}{T_S(N) + T_P(N)} \]

\[ W_S = \beta_x W_1, \text{ } W_1 \text{ is the power of one BCE} \]

\[ W_P = W_1 \sum_{i=1}^{X} \beta_i N_i = N_\beta W_1 \]

Idle power includes both static power and active power that’s not used by workload

\[ W_{idle} = N_i \cdot W_i \]

when \( N_i \) BCEs are idle
Effective power

• The $\beta$s are similar to the $\alpha$s, but pertain to power
  – An ith-type core consumes $\beta_i W_1$ power and has $\alpha_i$ speed (speed of one core is 1 – for throughput, we deal with speedup, for power, we deal with wattage and not ratios)
  – $N_\beta$ is the power-equivalent number of BCEs
  – For synchronizing on the slowest core:

\[
N_\beta = \min \bar{\alpha} \cdot \sum_{i=1}^{x} \frac{N_i \beta_i}{\alpha_i}
\]
Effective power

• Effective power formulas have been derived for all three types of models/laws

  – Can be viewed as results of ‘power scaling’ with PS functions:

\[
PS \left( \bar{N} \right) = \frac{\beta_X}{\alpha_X} \cdot (1 - P) + \frac{N_\beta}{N_\alpha} \cdot P
\]

\[
PS \left( \bar{N} \right) = \frac{\beta_X \cdot (1 - P) + N_\beta \cdot P}{\alpha_X \cdot (1 - P) + N_\alpha \cdot P}
\]

\[
PS \left( \bar{N} \right) = \frac{\beta_X \cdot (1 - P) + \frac{N_\beta}{N_\alpha} \cdot P \cdot g \left( \bar{N} \right)}{(1 - P) + P \cdot g \left( \bar{N} \right)}
\]

\[
W \left( \bar{N} \right) = PS \left( \bar{N} \right) \cdot SP \left( \bar{N} \right) \cdot W_1
\]

With \( \alpha_i = \beta_i = 1, \forall i \), and \( \bar{N} = N \)

all models transform to homogeneous forms
Efficiency

• Power-normalized performance
  – IPS/Watt
• Energy per instruction
  – Joules/Instruction
• Energy-normalized performance
  – IPS/Joule
• All models in the paper
Outline

• Existing speedup models
• Motivation
• Extended heterogeneous speedup models
• Power consumption models
• Power and energy normalized speedup
• Experimental results and cross validation
• Conclusions
# Experimental platform

<table>
<thead>
<tr>
<th>Exynos 5422 Application Processor</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Cortex-A15 Quad (2.0 GHz)</strong></td>
<td><strong>Cortex-A7 Quad (1.4 GHz)</strong></td>
</tr>
<tr>
<td><strong>Cortex-A15</strong></td>
<td><strong>Cortex-A7</strong></td>
</tr>
<tr>
<td>32 KB instruction cache</td>
<td>32 KB instruction cache</td>
</tr>
<tr>
<td>32 KB data cache</td>
<td>32 KB data cache</td>
</tr>
<tr>
<td>VFPv4</td>
<td>VFPv4</td>
</tr>
<tr>
<td><strong>Cortex-A15</strong></td>
<td><strong>Cortex-A7</strong></td>
</tr>
<tr>
<td>32 KB instruction cache</td>
<td>32 KB instruction cache</td>
</tr>
<tr>
<td>32 KB data cache</td>
<td>32 KB data cache</td>
</tr>
<tr>
<td>VFPv4</td>
<td>VFPv4</td>
</tr>
<tr>
<td><strong>2 MB Level 2 Cache with ECC</strong></td>
<td><strong>512 KB Level 2 Cache</strong></td>
</tr>
<tr>
<td><strong>GPU Mali-T628 MP6 (600 MHz)</strong></td>
<td><strong>DRAM LPDDR3 (933 MHz) 14.9 GBytes/s</strong></td>
</tr>
</tbody>
</table>
System characterization

- Build parameters through experimentation
  - \(W_{A7}, W_{A15}, W_{idle}\) (for the ‘whole’ – did not try differentiating different \(W_i\) – cores not turned off even in \(N_{A7}=0\) and \(N_{A15}=0\) cases)
  - \(\alpha_{A7}, \alpha_{A15}, \beta_{A7}, \beta_{A15}\)
  - May be different for different apps/computations
  - CPU-heavy tasks mainly experimented in this initial study, min \(\alpha\) scheduling
  - log, sqrt, and integer arithmetic tested
Exploration with models

• Run models with various execution scenarios to investigate the effects of P, DVFS, core scaling, etc. (large database available from technical report, some example data in the paper)

Exploration with models

- Run models with various execution scenarios to investigate the effects of P, DVFS, core scaling, etc. (large database available from technical report, some example data in the paper)

P=0.9

P=0.1
## Cross-validation

<table>
<thead>
<tr>
<th>bench</th>
<th>$P$</th>
<th>$N_{A7}$</th>
<th>$N_{A15}$</th>
<th>time, ms</th>
<th>speedup</th>
<th>average total power, W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>measured</td>
<td>predicted</td>
<td>measured</td>
<td></td>
<td>predicted</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.3</td>
<td>3</td>
<td>0</td>
<td>59992</td>
<td>1.2500</td>
<td>1.2505</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.3</td>
<td>0</td>
<td>4</td>
<td>61911</td>
<td>1.2116</td>
<td>1.2117</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.3</td>
<td>2</td>
<td>2</td>
<td>61910</td>
<td>1.2116</td>
<td>1.2118</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.3</td>
<td>3</td>
<td>4</td>
<td>59359</td>
<td>1.2641</td>
<td>1.2638</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.9</td>
<td>3</td>
<td>0</td>
<td>29988</td>
<td>2.5000</td>
<td>2.5017</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.9</td>
<td>0</td>
<td>4</td>
<td>25977</td>
<td>2.8893</td>
<td>2.8879</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.9</td>
<td>2</td>
<td>2</td>
<td>25961</td>
<td>2.8893</td>
<td>2.8897</td>
</tr>
<tr>
<td>sqrt</td>
<td>0.9</td>
<td>3</td>
<td>4</td>
<td>18300</td>
<td>4.1082</td>
<td>4.0995</td>
</tr>
<tr>
<td>int</td>
<td>0.9</td>
<td>3</td>
<td>0</td>
<td>31705</td>
<td>2.5000</td>
<td>2.5021</td>
</tr>
<tr>
<td>int</td>
<td>0.9</td>
<td>0</td>
<td>4</td>
<td>20823</td>
<td>3.8112</td>
<td>3.8097</td>
</tr>
<tr>
<td>int</td>
<td>0.9</td>
<td>2</td>
<td>2</td>
<td>24264</td>
<td>3.2708</td>
<td>3.2694</td>
</tr>
<tr>
<td>int</td>
<td>0.9</td>
<td>3</td>
<td>4</td>
<td>16637</td>
<td>4.7777</td>
<td>4.7682</td>
</tr>
<tr>
<td>log</td>
<td>0.9</td>
<td>3</td>
<td>0</td>
<td>25118</td>
<td>2.5000</td>
<td>2.5053</td>
</tr>
<tr>
<td>log</td>
<td>0.9</td>
<td>0</td>
<td>4</td>
<td>11580</td>
<td>5.4219</td>
<td>5.4341</td>
</tr>
<tr>
<td>log</td>
<td>0.9</td>
<td>2</td>
<td>2</td>
<td>17722</td>
<td>3.5492</td>
<td>3.5508</td>
</tr>
<tr>
<td>log</td>
<td>0.9</td>
<td>3</td>
<td>4</td>
<td>11690</td>
<td>5.3960</td>
<td>5.3830</td>
</tr>
</tbody>
</table>

Max 2.17%  
Max 0.24%
Outline

• Existing speedup models
• Motivation
• Extended heterogeneous speedup models
• Power consumption models
• Power and energy normalized speedup
• Experimental results and cross validation
• Conclusions
Conclusions

• Extended popular parallelization speedup models to cover a wider range of iso-ISA (or coefficient-equivalent ISA) heterogeneity
• Extended power models and efficiency models
• First cross-validation study successful
  • Need to investigate even wider scopes of heterogeneity
  • Need to study other speedup models (e.g. Downey’s)
  • Need to investigate realistic memory subsystems (cache misses)
  • Need to explore using these models for run-time management